

**CS1555 Database Management Systems**

**Final Project Report**

**Stefon Miller(SMM248)**

**Adam Buchinsky(ASB153)**

**Aaron Matthew(AAM150)**

User Manual

* **Administrator functions:**
  + Task 1-Erase the database
    - After selecting this function, you will have to provide confirmation in the format of “yes” or “y”. This will erase all tuples from all tables in the database.
  + Task 2-Add a customer
    - This function adds a new customer to the database. You have to supply the login, name, email, and password of said customer. The customer email must contain an ‘@’ symbol and a period. A balance can also be specified but is not required. By default the balance for a new customer is $0.
  + Task 3-Add new mutual fund
    - This function adds a new mutual fund to the database. You have to supply the symbol, name, description, and category of the fund. The fund category must be either “fixed”, “mixed”, “bonds”, or “stocks”. The system will add this fund to the database using the current date in the system
  + Task 4-Update Share quotes for day
    - This function reads in price data from a file and updates the closing prices of stocks in the database with symbols matching those in the file. If there are already closing prices for the current day, then the prices are simply updated with prices from the file. The system only accepts files with fund symbols and prices delimited by commas, with separate stocks on new lines.
  + Task 5-Show top-k highest volume categories
    - This function displays the top k fund categories based on the number of shares owned by customers. You must input a non-zero, positive integer for the top values. If the k value is higher than the total number of funds, the system will display all available funds.
  + Task 6-Rank all the investors
    - This function ranks all users in the database based on the total value of all of their shares. The value is calculated using the latest closing price.
  + Task 7-Update the current date(i.e., the pseudo date)
    - This function updates the system date to a specified date in the format “YYYY-MM-DD”. Invalid date formats are not accepted
* **Customer functions:**
  + Task 1 - Show the customer’s balance and total number of shares
    - This function displays the balance and total shares owned by the customer currently logged in.
  + Task 2 - Show mutual funds sorted by name
    - This function sorts and displays mutual funds alphabetically based on the mutual fund’s name(not the symbol).
  + Task 3 - Show mutual funds sorted by prices on a date
    - This function displays price information on mutual funds for a given date. You have to supply a specified date in the format “YYYY-MM-DD”. If the input date is before the system has records or after the current date in the system, it will display records from the first recorded and last recorded dates respectively. Funds owned by the currently logged in user will be denoted by a “\*\*”
  + Task 4 - Search for a mutual fund
    - This function returns mutual funds with descriptions matching 1 to 2 keywords. If 2 keywords are supplied, the system will only display the funds matching both keywords in the description. Keywords can contain commas, symbols, etc. Resulting funds will be displayed in the format “[symbol\_1, symbol\_2]”
  + Task 5 - Deposit an amount for investment
    - This function deposits a specified amount of money which is then invested in funds based on the user’s allocation preferences. If the amount deposited is not enough to buy all shares as specified in the user’s allocation preferences, then the money is simply deposited normally. If shares are bought, the remaining amount is the new balance in the account.
  + Task 6 - Buy shares
    - This function buys shares of a specified mutual fund. Share prices are calculated using the closing price for the most recent day. You can input either the exact number of shares to buy or an amount to use in the transaction. If the specified amount of shares costs more than the current user balance, no shares will be bought. If the user specifies an amount instead, the maximum number of shares for that amount will be bought and the remainder will be re-added to the user’s account.
  + Task 7 - This function sells shares of a specified mutual fund. Users should input the number of shares for the given mutual fund they wish to sell. If the user attempts to sell more shares than they own, they will not sell any shares. Once shares are sold, the resulting amount is added to the user’s balance.
  + Task 8 - Show ROI (return of investment)
    - This function displays the ROI for each fund purchased by the user. It calculates this using the current value of the investment - the initial cost over the initial cost.
  + Task 9 - Predict the gain or loss of the customer’s transactions
    - This function attempts to predict how much money the user will make based on their transactions. These predictions are based on the most recent closing prices of stocks owned by the user. The system also categorizes these transactions based on how successful the decisions were, “loss” refers to shares in which the current value is less than the price of the purchase transaction. “Profit” refers to shares in which the current value is more than the price of purchase. “Hold” refers to shares with prices equal to the purchase transaction.
  + Task 10 - Change allocation preference
    - This function changes allocation preferences of the user based on symbols entered by them. Users should also enter a corresponding percentage for the new preference. If the symbols already exist in the user’s allocation preferences, they will be updated instead. The total of all percentages must equal 100%. These preferences can be changed only once per day.
  + Task 11 - Rank the customer’s allocations
    - This function ranks the customer’s allocations based on ROI. The most recent closing price of all funds is used to calculate ROI.
  + Task 12 - Show portfolio [proc]
    - This function generates a custom portfolio report for the currently logged in user. It utilizes the shares currently owned by them for the portfolio. Users can see the funds they own, the number of shares owned, the current value of said funds, the price paid for the funds, and the total value of the portfolio based on prices for the current date.

System Limitations/Potential Improvements

The system assumes our team07.sql file is run previously in order to create the necessary tables, functions, procedures, views and triggers. Our system is not missing any functionality. All task specifications were met according to the project description. Our system would greatly improve from functionality for users to withdrawal funds or use them for their retirement. Additionally, having a more presentable interface would improve the user experience, but is not in the scope of this course. Our exception handling was a little messy and could be improved, but not without losing specific error messages for each task.